Porting the CFS V2 model suite to the NASA Ames SGI ICE platform (pleiades)

Larry Marx

Center for Ocean-Land-Atmosphere Studies

25 August 2011

TOPICS

- Timeline
- Program environment considerations
- Learning steps
- Installing model components and dependent libraries
- Installing post-processing program suite
- Ideas for CFS Version 3
- Conclusions

Timeline

- Preliminary code understanding and port to NCAR Bluefire running LSF: one month
- Installation of model libraries, component and support executables, debugging and testing: three months
- Installation of post-processing executable suite, debugging and testing: three months

Program environment considerations

- Key considerations:
  - Native big vs. little endian “byte gender”
  - Software availability
    - Fortran and C compiler: which vendor, which version?
    - Which Message Passing Interface (MPI) version works best with existing hardware and selected compilers
  - Other available software
    - NetCDF compatible with selected compilers and MPI
    - NetCDF operators to assist in manipulating and processing ocean model data
    - Library packages containing LAPACK and BLAS routines

Program environment considerations (cont.)

- Key considerations (cont.)
  - Batch environment
    - Local settings for MPI, working directories, etc.
    - Time, memory and processor limits
    - Dependencies with specific version of the operating system
    - Queues for testing and debugging
- Other considerations
  - Debug and performance tools
  - System support

Learning Steps

- Detailed understanding of model run scripts
  - Several thousand lines spanning many script files (up to 6 or 7 levels)
    - ➔ Can this be simplified?
  - Change scripts for local environment and batch system
- Determining which executables need to be built to prepare data in-stream and actually run the coupled model system.
- How can CFS v2 be changed to become a community model?

Installing model components and dependent libraries

- Build libraries for atmosphere model and related executables
- Build and test simple executables
- Build and test atmosphere model
- Build coupler and ocean model
- Test coupled system
- ➔ Problems:
  - Many versions of the same library
  - Some library versions lack big to little-endian conversion: merge required. Some outside NCEP versions needed.
  - Non-standard Fortran code and MPI usage
  - Ocean model (as modified) cannot run stand alone

Installing model components and dependent libraries (cont)

- ➔ Problems: (cont.)
  - Coupled system fails early in random locations and times in the ocean model. Not traceable via print statements. Solution: add more processors to the ocean model. Likely cause: lack of memory.
  - Some additional bugs uncovered while tracking down this problem
- Result:
  - Model completes one month in 75 minutes using 144 cores (vs 105 minutes for IBM running 128 cores)

Installing post-processing suite

- Similar steps to model installation with:
  - Additional libraries, including a full build of WRF
    - ➔ Why?
  - Many more executables needed, some not selected for installation by COLA due to lack of need.
  - Switch parallelization strategy from CPU based scaling to I/O based scaling.
- Addition of site specific data archival
- Post-processing and data archival usually completes in under 30 min. using a single 8-core node

Installation of post-processing suite (cont)

- Problem: due to I/O load and many sub-steps, a high probability of failure exists: a least one month per year can be expected to fail
- Solution: add more robust features to run script for recovery. Develop complicated recipe for typical failure recovery procedures (very messy).
- ➔ Are modern high performance disk systems unable to work with such a high I/O load? Can things be simplified?

Ideas for CFS Version 3

- Develop and test model components on many platforms and compilers at the same time
- Allow each model component to be tested stand alone
- Allow parts of each model to be switched on or off at execution time to aid in debugging and validation
- Provide workable low resolution data sets for testing
- Provide adequate documentation:
  - Scientific
  - Model installation
  - Model use
  - Technical, e.g. parallelization strategies and configuration

Ideas for CFS Version 3 (cont)

- Consider NetCDF (version 4 preferable) as an internal data standard. This allows better I/O performance, better scalability, use of more available tools including those developed at other centers. Distributed data can be converted to Grib2 with more complete meta-data.

Conclusions

- Porting CFS V2 to a new platform is a difficult labor intensive task
- Substantial revisions are needed to the full NCEP software infrastructure in order to improve the portability and utility to the outside community.
- These changes can result in a more sustainable software environment and provide a larger pool of well qualified new staff members for NCEP.